

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-032206

(43)Date of publication of application : 02.02.1999

(51)Int.Cl.

H04N 1/387

G06T 11/60

H04N 1/40

(21)Application number : 09-185315

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 10.07.1997

(72)Inventor : MANJIYUUME YUMI  
SEKINE HIROSHI

(54) IMAGE PROCESSOR

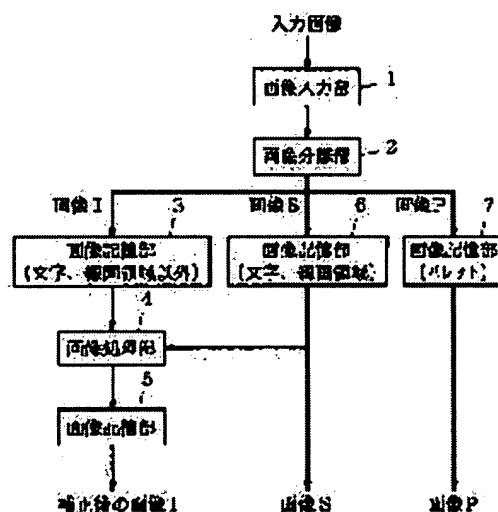
(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide an image processor which can suppress the occurrence of voiding or color image deterioration at the time of compositing images and can image enhance compression efficiency at the time of performing separation and compression.

**SOLUTION:** In an image processor, an image separating section 2 separates an image inputted to an image inputting section 1 into an image S indicating the state of a character or a line drawing, an image P indicating the color information of the character or line drawing, and an image I which is obtained by deleting the data of characters and of line drawings from the inputted image.

An image-storing section 3 temporarily stores an image I in which the existing area of the characters or line

drawings is voided. In the image element in the voided area of the image I, an image processing section 4 fills up the voided area with an image element value by referring to the image element value in the periphery of the voided area. Thus, even the existing areas of the characters and line drawings in the image also become flat image, and the characters and line drawings can be composited with each other without causing voids, and compression efficiency can be enhanced.



## LEGAL STATUS

[Date of request for examination] 16.08.2001

[Date of sending the examiner's decision of rejection] 10.08.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-32206

(43) 公開日 平成11年(1999) 2月2日

(51) Int.Cl.<sup>8</sup>

識別記号

F I

H 0 4 N 1/387

H 0 4 N 1/387

G 0 6 T 11/60

G 0 6 F 15/62

3 2 5 P

H 0 4 N 1/40

H 0 4 N 1/40

F

審査請求 未請求 請求項の数 3 O L (全 13 頁)

(21) 出願番号 特願平9-185315

(22) 出願日 平成9年(1997) 7月10日

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 万城目 ゆみ

神奈川県海老名市本郷2274番地 富士ゼロックス株式会社内

(72) 発明者 関根 弘

神奈川県海老名市本郷2274番地 富士ゼロックス株式会社内

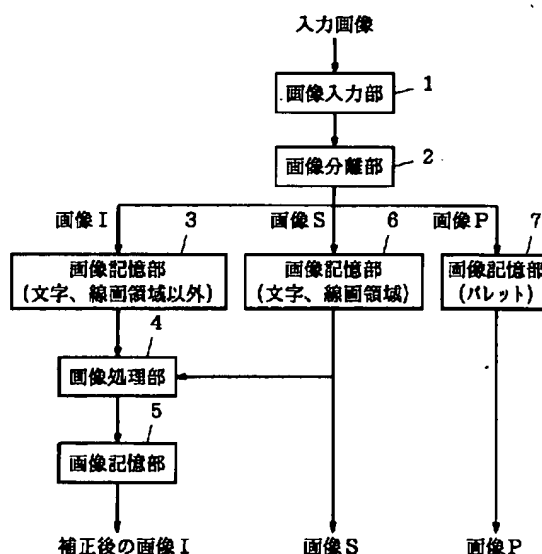
(74) 代理人 弁理士 石井 康夫 (外1名)

(54) 【発明の名称】 画像処理装置

(57) 【要約】

【課題】 合成時の白抜けや画像品質の劣化を抑え、また、分離圧縮時の圧縮効率を高めた画像処理装置を提供する。

【解決手段】 画像入力部1に入力された画像を、画像分離部2において文字または線画の形状を示す画像Sと、文字または線画の色情報を示す画像Pと、入力画像から文字または線画のデータを削除した画像Iに分離する。画像Iは画像記憶部3に一時保持される。画像Iは、文字または線画が存在していた部分が白抜けとなっている。画像処理部4において、画像Iの白抜け領域の画素について、白抜け領域の周辺の画素値を参照して画素値を埋め込む処理を行なう。これによって画像Iの文字や線画が存在していた部分も平坦な画像となり、文字・線画との合成時に白抜けは発生せず、圧縮効率を高めることができる。



## 【特許請求の範囲】

【請求項1】 入力された画像情報を文字または線画情報からなる第1の画像情報と文字または線画情報以外の情報からなる第2の画像情報に分離する分離手段と、該分離手段により分離された前記第2の画像情報について前記第1の画像情報として分離された領域の一部または全部に対し当該領域の周囲の画素を参照して画像情報を埋め込む画像情報埋め込み手段を有することを特徴とする画像処理装置。

【請求項2】 前記画像情報埋め込み手段は、前記領域の注目画素における縦方向または横方向における同一ラインに位置する前記周辺画素を参照して該周辺画素の画素値に基づいて前記注目画素の画素値を決定することを特徴とする請求項1に記載の画像処理装置。

【請求項3】 前記画像情報埋め込み手段は、前記画素の注目画素に対する前記周辺画素の距離および前記周辺画素の画素値に基づいて前記注目画素の画素値を決定することを特徴とする請求項1に記載の画像処理装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、入力された画像情報を複数の画像情報に分離して処理を行なう画像処理装置に関するものである。

## 【0002】

【従来の技術】入力された画像を保存や転送する場合には、なるべくデータ量を減少させることが望まれる。データ量を削減するために、入力画像を例えば文字や線画情報と、それ以外の1以上の画像情報に分離し、それぞれ最適なデータ圧縮処理が施される場合がある。また、データ圧縮処理前に、分離された画像情報毎に異なる解像度へ画像を変換し、データ量をさらに削減する場合もある。

【0003】特開平4-105178号公報では、入力画像から文字領域と図表領域とイメージ領域を分離抽出し、各領域に含まれる情報に適するデータに変換後、再構築する構成が開示されている。

【0004】図16は、従来の画像処理装置における画像分離処理の具体例の説明図である。図16(A)は入力画像であり、ハッチングを施して示した部分はイメージであり、イメージの上に文字「ABC」が重なっている。このような入力画像に対して画像分離処理を行なうと、図16(B)に示すような文字領域を表わす画像と、図16(B)に示すようなイメージ領域を表わす画像に分離される。上述の文献では、分離された文字領域について文字認識処理を行ない、データ量を削減している。

【0005】一般にイメージ部分の画像は、データ量が多い。例えば600dpiのスキャナからA4サイズの画像をRGB各色8ビットで読み取った場合のデータ量は、約108Mバイトになってしまう。現状では一般

に文字や線画部分の解像度は600dpiまたはこれ以上の高解像度が必要とされているが、イメージ部分の解像度は文字や線画部分と比べて低解像度でも差し支えない。そのため、イメージ部分の解像度を文字部分よりも下げたり、JPEGベースライン方式に代表されるような圧縮手法を用いてデータ量を削減し、データの保持や通信が行なわれている。

【0006】例えば図16(A)に示す入力画像からイメージ部分のみピックアップした図16(C)に示す画像を低解像度で保持する場合を考える。出力する際には低解像度で保持している画像を元の解像度へ解像度変換し、図16(B)に示す文字部画像と合成する。このとき、イメージ部分を元の解像度へ変換した際に図16(C)に示す画像とエッジ部分で相違することがあり、文字部分とイメージ部分の境界部に白抜けが発生する可能性がある。

【0007】また、図16(C)に示すように、イメージ領域のみを分離した画像上では、文字が存在していた領域が白抜けする。特にイメージ上に細かい文字が多く書かれている場合には、イメージ領域のみ分離した画像上で白抜けが頻繁に存在し、画像中の高周波成分が大きくなる。このような画像を、JPEGベースライン方式等の離散コサイン変換を用いた可変長かつ非可逆データ圧縮方式を用いて圧縮すると、圧縮率の低下や、圧縮伸長後の画像品質の劣化するという問題もある。

## 【0008】

【発明が解決しようとする課題】本発明は、上述した事情に鑑みてなされたもので、合成時の白抜けや画像品質の劣化を抑え、また、分離圧縮時の圧縮効率を高めた画像処理装置を提供することを目的とするものである。

## 【0009】

【課題を解決するための手段】本発明は、入力された画像情報を文字または線画情報からなる第1の画像情報と、文字または線画情報以外の情報からなる第2の画像情報に分離し、分離された第2の画像情報については、第1の画像情報として分離された領域の一部または全部に対して、当該領域の周囲の画素を参照して画像情報を埋め込む。これによって第2の画像情報を低解像度に変換し、再び元の解像度に戻して第1の画像情報と合成しても、埋め込まれた画像情報によって白抜けを起こすことはない。また、イメージ部分に文字や線画が存在していても、第2の画像情報では文字や線画の部分に周囲の画素を参照して、例えば圧縮効率が高くなるように画像情報を埋め込むことができる。そのため、文字や線画が存在していた部分で第2の画像情報の圧縮効率が低下することはなく、効率よく圧縮することができるとともに、復元時の画像品質を維持することができる。

## 【0010】

【発明の実施の形態】図1は、本発明の画像処理装置の実施の一形態を示すブロック図である。図中、1は画像

入力部、2は画像分離部、3、5～7は画像記憶部、4は画像処理部である。画像入力部1は、イメージスキャナ等の画像入力装置あるいはネットワークから画像を入力するインターフェース部などによって構成され、入力された画像を画像分離部2に渡す。入力された画像は、カラー画像や濃淡画像などであってよい。

【0011】画像分離部2は、入力された画像を文字または線画の領域とそれ以外の領域に分離する。この例では、文字または線画の領域については、「文字または線画の形状を示すデータ」と、「文字または線画の色情報を示すパレットデータ」の2つのデータに分離する。また文字または線画の領域以外の領域の画像とは、入力画像から「文字または線画のデータを削除したデータ」である。以下、「文字または線画の形状を示すデータ」を画像S、「文字または線画の色情報を示すパレットデータ」を画像P、「文字または線画のデータを削除したデータ」を画像Iと略す。分離した画像Iは画像記憶部3へ、画像Sは画像記憶部6へ、画像Pは画像記憶部7へそれぞれ格納する。この画像分離部2における画像の分離方法は、例えば特開平3-126180号公報や、上述の特開平4-105178号公報に記載されている方式など、多くの方式が提案されており、どのような方式を採用するかは任意である。

【0012】画像処理部4は、画像記憶部3に格納されている画像I中の文字または線画として抽出されて抜けている部分に対し、画像記憶部6に格納されている画像Sを用いて補正処理を施し、画像記憶部5に格納する。

【0013】画像記憶部5は、画像処理部4で補正処理された画像Iが格納される。画像記憶部5、6、7に格納されている補正処理された画像I、画像S、画像Pは、それぞれ出力される。出力された各画像は、例えば解像度変換したり符号化して、データベースに格納したり、あるいはネットワークなどによって通信したりすることができる。解像度変換や符号化の際には、それぞれの画像について、それぞれの画像の特性に応じた解像度に変換したり、それぞれの画像の特性に応じた符号化方式で符号化すればよい。一例としては、画像Iは解像度を下げてJPEGベースライン方式等により符号化して格納することができる。また、画像Sはそのままの解像度で例えばJBI方式などによって符号化して格納することができる。画像Pは、非常に解像度を下げて、例えばJPEGベースライン方式等によって符号化して格納することができる。

【0014】以下、画像処理部4における処理動作の例について説明する。図2は、画像データの説明図である。以下の説明では画像を2次元の座標で扱い、図2に示すように画像の左上の座標を $(i, j) = (1, 1)$ 、右上の座標を $(i, j) = (i_{\max}, 1)$ 、左下の座標を $(i, j) = (1, j_{\max})$ 、右下の座標を $(i, j) = (i_{\max}, j_{\max})$ とする。

【0015】また以下の説明において、 $S(i, j)$ は画像Sの座標 $(i, j)$ に対応する画素値、 $I(i, j)$ は画像Iの座標 $(i, j)$ に対応する画素値を表わすものとする。ここで、画像Sは1ビットのデータであり、各画素値 $S(i, j) = 1$ は「文字または線画」であることを示し、 $S(i, j) = 0$ は当該画素が「文字または線画」以外であることを表わす。また、画像Iは階調を持ったデータであり、カラーの場合には画素値 $I(i, j)$ は例えば「赤(R)、青(B)、緑(G)」のような複数の色に分色されたデータをそれぞれ多値データ(例えば8ビットで256階調)で表わされる。以下の説明では一つの色成分について示すが、複数の色成分に分色されている場合には、それぞれの色に対して同じ演算を施せばよい。

【0016】図3、図4は、画像処理部の第1の動作例を示すフローチャートである。ここでは、画像処理部4は、画像I中の文字または線画として抽出されて抜けている部分に対し、その部分の両端部の画素値を用いて埋め込む例を示している。より具体的には、各ライン(各j)ごとに画像Sを参照し、画素値が0から1に変わる点を検出して文字や線画の存在する部分の左端(開始点)とする。同様に、画像Sの画素値が1から0に変わる点を検出して右端(終了点)を求める。検出された開始点から終了点までが画像Iにおける白抜け部分である。この白抜け部分を埋める処理として、この例では、その中央より左側では左端部の画像Iの画素値により埋め、中央より右側では右端部の画像Iの画素値により埋める。これによって画像I中の文字または線画として抽出されて白抜け部分を各ラインごとに埋めてゆく。

【0017】まずS21において、処理を行なうラインjを0として初期化し、次にS22において、jに1を加えて次のラインを処理ラインとする。処理開始直後ではj=0に1を加え、j=1となって先頭ラインを示すようになる。またS22では処理ラインj中の処理すべき列iを0として初期化し、S23においてiに1を加え、次の画素を注目画素とする。各ラインの先頭では、i=0に1を加え、i=1となってラインの先頭画素を示すようになる。

【0018】S23～S29では、画像Iにおける白抜け領域の開始点を検出する。S23でiを更新後、S24において画像Sを参照して開始点か否かを判定する。すなわち、画像Sにおける注目画素に対応する位置の画素値 $S(i, j)$ が0で、次の画素値 $S(i+1, j)$ が1か否かを検出する。この条件を満足する場合、文字や線画領域の開始点であるものとして、S47において変数 $\alpha$ に画像Iにおいて対応する座標を持つ画素値 $I(i, j)$ を格納し、変数Aにその座標iを格納してS30に進む。変数 $\alpha$ は、白抜け領域の開始点における画素値を保存し、変数Aは開始点の座標を保存する。

【0019】また、S24の条件を満足しない場合、ラ

インの先頭から白抜け領域となっている場合も考えられるので、S25において $i=1$ 、すなわちラインの先頭か否かを判定して、ラインの先頭の場合にはS28において画像Sの画素値 $S(i, j)$ が1か否かを判定する。ラインの先頭で画像Sの画素値 $S(i, j)$ が1の場合には、ラインの先頭から白抜け領域となっている。この場合にはS29において埋めるべき画素値として変数 $\alpha$ に0を、開始点の座標として変数Aに1を格納し、S30に進む。

【0020】S25において注目画素がラインの先頭でない場合には、S26において注目画素がラインの右端の1画素手前まで達したか否かを判定し、右端の1画素手前まで達していれば、次のラインの処理に移行すべくS39へ進む。なお、右端の画素については次の画素が存在しないのでS24における判定を行なう必要はなく、開始点の検出の処理は行なわれない。S24で開始点を検出されず、さらにS26において注目画素がラインの右端の2画素以上手前である場合、またはS25においてラインの開始点と判定されたがS28において開始点が文字や線画でなかった場合には、S23へ戻り、 $i$ に1を加えて注目画素を次の画素に移す。

【0021】このような処理を繰り返してゆく過程で、ライン $j$ 上の白抜け部分の開始点（ラインの先頭から始まる場合を含む）が検出された場合には、S30へ進む。S30～S34では白抜け領域の終了点を検出する処理である。S30において $i$ に1を加えて注目画素を次の画素に移した後、S31において、画像Sにおける注目画素に対応する位置の画素値 $S(i, j)$ が1で、次の画素値 $S(i+1, j)$ が0か否かを判定する。条件を満たす場合、S33において変数 $\beta$ に画像Iにおいて対応する座標を持つ画素値 $I(i+1, j)$ を格納し、変数Bにはその座標 $i+1$ を代入して、S35に進む。変数 $\beta$ は、白抜け領域の終了点における画素値を保存し、変数Bは終了点の座標を保存する。

【0022】S31において条件を満たさない場合には、S32において注目画素が右端の手前の画素か否かを判定する。すなわち、白抜け領域がラインの右端まで連続している場合を判定する。白抜け領域がラインの右端まで達している場合には、S34において変数 $\beta$ に画素値として0を格納し、変数Bに終了点の座標として $i_{\max}$ を格納する。S31で終了点を検出されず、注目画素がラインの右端の2画素以上手前である場合には、S30へ戻り、 $i$ に1を加えて次の画素を注目画素とする。

【0023】このような処理を繰り返すことによって、S23～S29の処理で白抜け領域の開始点を検出された場合のその白抜け領域の終了点を検出する。そしてS35へ進む。

【0024】S35～S37は、白抜け領域に保存しておいた画素値を埋め込む処理を行なう。まずS35にお

いて、白抜け領域の開始点における画素値（変数 $\alpha$ の値）と終了点における画素値（変数 $\beta$ の値）を比較し、同じ値であればS37において画像Iの画素値 $I(A, j) \sim I(B, j)$ までを一律に変数 $\alpha$ または変数 $\beta$ の値とする。また、変数 $\alpha$ と変数 $\beta$ の値が異なる場合には、S36において、画像Iの画素値 $I(A, j) \sim I(A + [(B-A)/2], j)$ を変数 $\alpha$ の値とし、画素値 $I(A + [(B-A)/2] + 1, j) \sim I(B, j)$ を変数 $\beta$ の値とする。ここで、 $[\ ]$ はガウス記号を表わしている。このようにして白抜け領域はその開始点または終了点の画素値によって埋められる。

【0025】S38において注目画素がラインの右端まで達したか否か、すなわち1ラインの処理を全て終えたか否かを判定し、ラインの途中である場合にはS23へ戻り、白抜け領域の開始点の検出を続ける。また、1ラインの処理を終えた場合には、S39において処理ラインが最終ラインであったか否かを判定し、最終ラインでない場合にはS22へ戻り、次のラインを処理ラインとして処理を続ける。最終ラインまで処理を繰り返して行ない、最終ラインの処理が終了した場合は画像処理部4における処理を終了する。

【0026】上述の例では白抜け領域の開始点と終了点における画素値が異なる場合に、中央点 $C = A + [(B-A)/2]$ を境としてその左側を開始点における画素値、その右側を終了点における画素値とした。しかしこれに限らず、境界とする点は開始点、終了点に依存するものであれば任意に決めることができる。もちろん、開始点あるいは終了点の画素値のいずれかで白抜け領域すべてを埋めてもよい。

【0027】また、上述の第1の動作例ではライン毎に処理を行なうため、必要とするメモリ量は少なくない。さらに、1画素毎ではなく連続した白抜け部分の画素値を一度に変換するために、処理時間が短縮できる。しかも、処理後の画像Iにおいては、文字や線画の存在していた部分において開始点や終了点から同じ画素値が連続するため、例えばラインごとに変化点を検出するような圧縮方法を用いて画像Iを符号化した場合、符号化効率を向上させることができる。

【0028】上述の第1の動作例を、具体例を用いて説明する。図5は、入力画像および分離された各画像の具体例の説明図である。図5(A)は入力画像の具体例を示しており、写真などの中間調画像上に「ABC」の3文字が書かれている画像である。中間調部分にはその濃度に応じたハッチングを施して示している。このような入力画像は、画像分離部2によって図5(B)に示す画像I、図5(C)に示す画像S、図5(D)に示す画像Pに分離される。この状態では画像Iにおいて文字「ABC」が書かれていた部分が図5(B)に示すように白抜けの状態となっている。

【0029】図6は、分離された各画像および処理後の

画像Iにおける画素値の具体例の説明図である。図5(B)および図5(C)における横線11上の画素値の一部を図6(B)および図6(A)に示している。なお、図6では説明を簡単にするため非常に低解像度の場合を示している。図6(B)に示す画像Iにおいて、画素値は8ビットで「0」を白、「255」を黒とし、図5において右上がりのハッチングを施した部分を画素値20、クロスハッチングを施した部分を画素値50、境界を画素値90としている。画像分離部2によって分離した結果、図5(B)に示す画像Iにおいて文字「ABC」が削除された部分は、図6(B)に示すように画素値が0になっている。また、文字や線画として分離された画像Sにおいては、図6(A)に示すように、文字「ABC」が存在している部分についてのみ1となっている。なおこの例では、文字「ABC」の濃度等は画像Pに分離されている。

【0030】いま、処理ラインjが図5(B)および図5(C)において示した横線11のラインであるものとする。図3、図4に示したフローチャートに従い、このラインの左端から順に注目画素を右方向にずらしながら処理を進めてゆく。i=pのとき、図6(A)からS(p, j)=0、S(p+1, j)=1であるのでS24の条件が成立し、画像Iにおける開始点が検出される。S27において変数αにI(p, j)=20が格納され、変数Aにpが格納される。

【0031】次にi=qにおいてS(q, j)=1、S(q+1, j)=0となり、S31の条件が成立する。これによって画像Iの白抜け部分の終了点を検出し、S33において変数βにI(q+1, j)=20、変数Bにq+1が格納される。

【0032】そして、変数Aに格納されている位置pから変数Bに格納されている位置q+1までの白抜け部分を埋める。S35において、変数αと変数βの値を比較すると、値は等しいのでS37において白抜け部分を変数αまたは変数βの値20で埋める。これによって図6(C)に示すように位置pからq+1までの白抜け領域が値20となり、白抜け領域であった部分を含め、値20の部分が連続することになる。

【0033】同様にして位置rからs+1までの領域、位置tからu+1までの領域、位置vからw+1までの領域の各白抜け部分では、その開始点あるいは終了点の画素値50によって埋められる。さらに、位置xからy+1までの白抜け領域では、開始点の画素値αが50、終了点の画素値βが20であり、αとβが等しくない。そのため、S36において、位置x+1, x+2については画素値50、位置y-1, yについては画素値20が埋められる。

$$I(i, j) = \{(i-A)\beta + (B-i)\alpha\} / (B-A) \quad \cdots (1)$$

画素値I(A, j)~I(B, j)をこの(1)式によって変換した後はS38へ進み、終了するまで処理を続

【0034】このようにして、図6(C)に示すような一連の画素値が得られる。これは図6(B)において存在していた白抜け部分が埋められた画像を示している。さらに図6(C)に示すように、白抜け部分を埋めることによってその周囲とも画素値が連続し、同じ画素値を有する区間が長くなっている。これによって例えばMH符号化等のようにランレングスを用いた符号化や、ラインごとの予測符号化を行なう場合には、ラン長が長くなりまた予測が的中しやすくなるため、符号化効率を向上させることができる。

【0035】また、図5(B)に示すように白抜け部分を有する画像Iを、データ量を減少させるために低解像度の画像に解像度変換して保存しておき、出力時に元の解像度に戻すと、白抜け部分のエッジが例えざざざざになるなどして、もとの形状に戻らない場合がある。このような状態で図5(C)および図5(D)に示す画像Sおよび画像Pを用いて合成すると、文字の周囲に白抜けが発生する恐れがある。しかし、本発明では白抜け部分を埋めているので、文字との境界は画像Iには存在しなくなり、解像度変換による影響は受けない。そのため、合成後の画像に白抜けが発生する恐れはなくなり、良好な画像を出力することが可能となる。

【0036】さらに、この第1の動作例ではライン毎の処理を行なうため、必要とするメモリが非常に少なく済み、また埋め込む画素値を周辺画素と同じ画素値とすることによって処理が簡単であり、高速な処理が可能である。

【0037】図7、図8は、画像処理部の第2の動作例を示すフローチャートである。上述の第1の動作例では、白抜け部分を埋める際に、白抜け部分の開始点および終了点の画素値をそのまま用いて埋めた。この第2の動作例では、白抜け部分の開始点および終了点の画素値から、新たに白抜け部分の画素値を作り出す場合を示す。

【0038】図7、図8において、図3、図4と同様の処理を行なう部分には同じ符号を付してあり、その部分の説明を省略する。上述の第1の動作例と異なるのは、S36の処理がS41の処理に変更されている点のみである。S35に至るまでの処理によって、白抜け部分の開始点の座標とその画素値および終了点の座標とその画素値が得られている。S35において開始点の画素値αと終了点の画素値βが等しくない場合、S41において白抜け部分の画素値を生成する。

【0039】S41では、白抜け部分の画素値すなわち画素値I(A, j)~I(B, j)を、それぞれ開始点Aと終了点Bからの距離に応じて次のように決定する。

【0040】図9は、画像処理部の第2の動作例におけ

る画像Iの画素値の具体例の説明図である。上述の第1の動作例と同様に、図5(A)に示す入力画像が図5(B)～(D)に示す各画像に分離されたものとする。図9(A)、(B)は図6(A)、(B)と同じものであり、図5(C)、(B)に示す画像S、画像Iの横線11上の一部の画素値の具体例を示している。この第2の動作例では、特に一点鎖線で囲んだ部分が上述の第1の動作例と異なる動作を行なう部分である。

【0041】図7、図8に示したフローチャートに沿って具体例について動作を説明するが、特に上述の第1の動作例と異なる部分のみについて説明する。位置 $p \sim q+1$ 、位置 $r \sim s+1$ 、位置 $t \sim u+1$ 、位置 $v \sim w+1$ の各白抜け領域については、その開始点および終了点の画素値が等しいので、S37において開始点あるいは終了点の画素値で白抜け領域が埋められる。

【0042】位置 $x$ において開始点が検出され、その位置が変数Aに格納されるとともに画素値50が変数 $\alpha$ に格納される。また、位置 $y$ において終了点が検出され、その位置 $y+1$ が変数Bに格納されるとともに画素値20が変数 $\beta$ に格納される。S35において変数 $\alpha$ と変数 $\beta$ の値を比べると異なるので、S41において白抜け領域を埋める処理が行なわれる。例えば $I(x+1, j)$ は上述の(1)式より、

$(1 \cdot \beta + 4 \cdot \alpha) / 5 = (20 + 200) / 5 = 44$ となり、画素値44が埋め込まれる。同様にして位置 $y$ の画素まで埋められる。このようにして画素が埋められた結果を図9(C)に示している。位置 $x$ から位置 $y+1$ までの間の画素値が段階的に変化していることがわかる。

【0043】このように、白抜け領域の画素を段階的に変化させて埋め込むことによって、階調の急激な変動を少なくし、高周波成分の発生を抑えることができる。そのため、例えば離散コサイン変換等の周波数成分への変換を伴う符号化方式を用いる場合においては、符号量を低減し、また復号された画像の品質劣化を抑えることができる。

【0044】もちろん、画像Iにおいて文字や線画として分離されて白抜けとなっている部分が埋められているので、画像Iを低解像度に変換後、元の解像度に再変換して文字や線画と合成しても、文字や線画のエッジ部分に白抜けは発生せず、良好な合成画像を得ることができる。また、この第2の動作例もライン毎の処理を行なうため、必要とするメモリが非常に少なく済むという利点がある。

【0045】次に画像処理部4の第3の動作例について説明する。上述の第1および第2の動作例では、画像Iにおいて白抜けした部分を埋める際に、同じライン上にある周辺2画素を参照し、白抜け部分の画素値を決定した。この第3の動作例では、同じライン上の周辺2画素とともに、同じカラム上の周辺2画素を参照し、合計4

画素の画素値から新たに白抜け領域の各画素値を作り出す例を示している。

【0046】図10ないし図12は、画像処理部の第3の動作例を示すフローチャートである。S51～S53は図3におけるS21～S23と同様の処理であり、まずS21において $j=0$ として初期化し、次にS22において、 $j$ に1を加えて次のラインを処理ラインとする。またS22では $i=0$ として初期化し、S23において $i$ に1を加え、次の画素を注目画素とする。

【0047】S53～S59は図3におけるS23～S29と同様の処理であり、画像Iにおける白抜け領域の開始点を検出する。S53で $i$ を更新後、S54において画像Sを参照して開始点が否かを判定する。すなわち、画像Sにおける注目画素に対応する位置の画素値 $S(i, j)$ が0で、次の画素値 $S(i+1, j)$ が1か否かを検出する。この条件を満足する場合、文字や線画領域の開始点であるものとして、S57において変数 $\alpha$ に画像Iにおいて対応する座標を持つ画素値 $I(i, j)$ を格納し、変数A1にその座標 $i$ を格納する。さらに、このときの座標 $j$ を変数Cに格納し、S60に進む。変数 $\alpha$ 1は、白抜け領域のライン方向の開始点における画素値を保存し、変数A1にはライン方向の開始点の座標を保存する。

【0048】また、S54の条件を満足しない場合、ラインの先頭から白抜け領域となっている場合も考えられるので、S55において $i=1$ 、すなわちラインの先頭か否かを判定して、ラインの先頭の場合にはS58において画像Sの画素値 $S(i, j)$ が1か否かを判定する。ラインの先頭で画像Sの画素値 $S(i, j)$ が1の場合には、ラインの先頭から白抜け領域となっている。この場合にはS59においてライン方向の開始点の画素値として変数 $\alpha$ 1に0を、ライン方向の開始点の座標として変数A1に1を格納する。さらに、このときの座標 $j$ を変数Cに格納し、S60に進む。

【0049】S55において注目画素がラインの先頭でない場合には、S56において注目画素がラインの右端の1画素手前まで達したか否かを判定し、右端の1画素手前まで達していれば、次のラインの処理に移行すべくS82へ進む。なお、右端の画素については次の画素が存在しないのでS54における判定を行なう必要はなく、開始点の検出の処理は行なわない。S54で開始点が出されず、さらにS56において注目画素がラインの右端の2画素以上手前である場合、またはS55においてラインの開始点と判定されたがS58において開始点が文字や線画でなかった場合には、S53へ戻り、 $i$ に1を加えて注目画素を次の画素に移す。

【0050】このような処理を繰り返してゆく過程で、ライン $j$ 上の白抜け部分の開始点が検出できる。開始点が出された場合には、S60へ進む。S60～S64では、図3のS30～S34と同様に、白抜け領域のラ

イン方向の終了点を検出する。S60において $i$ に1を加えて注目画素を次の画素に移した後、S61において、画像Sにおける注目画素に対応する位置の画素値 $S(i, j)$ が1で、次の画素値 $S(i+1, j)$ が0か否かを判定する。条件を満たす場合、S63において変数 $\beta 1$ 画像Iにおいて対応する座標を持つ画素値 $I(i+1, j)$ を格納し、変数 $B 1$ はその座標 $i+1$ を代入して、S65に進む。変数 $\beta 1$ は、白抜け領域のライン方向の終了点における画素値を保存し、変数 $B 1$ はライン方向の終了点の座標を保存する。

【0051】S61において条件を満たさない場合には、S62において注目画素が右端の手前の画素か否かを判定する。すなわち、白抜け領域がラインの右端まで連続している場合を判定する。白抜け領域がラインの右端まで達している場合には、S64において変数 $\beta 1$ に画素値として0を格納し、変数 $B 1$ に終了点の座標として $i_{\max}$ を格納する。S61で終了点が見つでられず、注目画素がラインの右端の2画素以上手前である場合には、S60へ戻り、 $i$ に1を加えて次の画素を注目画素とする。

【0052】このような処理を繰り返すことによって、S53～S59の処理で白抜け領域のライン方向の開始点が見つでされた場合のその白抜け領域のライン方向の終了点を検出することができる。そしてS65へ進む。

【0053】S65～S80では、上述のようにして検出した白抜け領域の開始点の次の画素から終了点の手前の画素までについて、それぞれの画素値を算出して埋め込む処理を行なう。S65において $i$ を白抜け領域のライン方向の開始点である変数 $A 1$ にセットし、S66において、 $i$ に1を加えて画素値を埋め込むべき画素を次の画素に設定する。例えばS65で $i$ を変数 $A 1$ にセットした直後では、白抜け領域の開始点の次の画素となる。そしてS67～S79で画素値を算出して埋め込み、S80で座標 $i$ が白抜け領域のライン方向の終了点 $B 1$ の手前の画素の座標であるか否か、すなわち白抜け領域のライン方向のすべての画素について埋め込み処理が終了したか否かを判定し、未処理の画素が残っている場合にはS66へ戻って座標 $i$ に1を加え、画素値を埋め込むべき画素として次の画素を設定して処理を繰り返す。

【0054】S67～S79において埋め込むべき画素値を算出して埋め込む処理を行なうが、この動作例では同じカラム上の白抜け領域の開始点および終了点の2画素の画素値も参照する。そのために、S67～S71ではカラム方向の白抜け領域の開始点を求め、S72～S77ではカラム方向の白抜け領域の終了点を求める。

【0055】S67～S71では、画素値を埋め込むべき画素についてカラム方向の座標 $j$ を減少させてゆき、カラム方向の開始点を検出する。S67において $j$ から1を減じ、S68においてライン $j$ の同じカラム位置に

おける画像Sの画素値 $S(i, j)$ が0で、次のライン $j+1$ の同じカラム位置における画像Sの画素値 $S(i, j+1)$ が1か否かを判定する。条件を満たす場合、S70において変数 $\alpha 2$ に画像Iにおいて対応する座標を持つ画素値 $I(i, j)$ を格納し、変数 $A 2$ にはその座標 $j$ を代入して、S72に進む。変数 $\alpha 2$ は、白抜け領域のカラム方向の開始点における画素値を保存し、変数 $A 2$ はカラム方向の開始点の座標を保存する。

【0056】S68において条件を満たさない場合には、S69において $j=1$ か否か、すなわち上端のラインまで達したか否かを判定する。この判定で、白抜け領域が上端のラインまで連続している場合を判定する。白抜け領域が上端のラインまで達している場合には、S71において変数 $\alpha 2$ に画素値として0を格納し、変数 $A 2$ に開始点の座標として1を格納する。S68で終了点が見つでられず、また上端のラインまで達していない場合には、S67へ戻り、 $j$ から1を減じて1ライン前の同じカラム位置について、白抜け領域のカラム方向の開始点か否かを判定する。このような処理を繰り返すことによって、カラム $i$ における画素値を埋め込むべき画素に対する白抜け領域のカラム方向の開始点を求めることができる。

【0057】同様に、S72～S77においてカラム $i$ における白抜け領域のカラム方向の終了点を求める。S72において $j$ をもとの処理ラインの手前のライン $(C-1)$ にセットし、S73において $j$ に1を加える。S74において、ライン $j$ における画像Sの画素値 $S(i, j)$ が1で、次のライン $j+1$ の画像Sの画素値 $S(i, j+1)$ が0か否かを判定する。条件を満たす場合、S76において変数 $\beta 2$ に画像Iにおいて対応する座標を持つ画素値 $I(i, j+1)$ を格納し、変数 $B 2$ にはその座標 $j+1$ を代入して、S78に進む。変数 $\beta 2$ は、白抜け領域のカラム方向の終了点における画素値を保存し、変数 $B 2$ はカラム方向の終了点の座標を保存する。

【0058】S74において条件を満たさない場合には、S75において最終ラインまで達したか否かを判定する。すなわち、白抜け領域が最終ラインまで連続している場合を判定する。白抜け領域が最終ラインまで達している場合には、S77において変数 $\beta 2$ に画素値として0を格納し、変数 $B 2$ に終了点の座標として $j_{\max}$ を格納する。S74で終了点が見つでられず、最終ラインでもない場合には、S73へ戻り、 $j$ に1を加えて次のラインの同じカラムの画素について判定を行なう。このような処理を繰り返すことによって、カラム $i$ における白抜け領域の終了点を検出することができる。

【0059】次にS78においてライン $j$ を変数 $C$ に保存しておいた座標に戻し、S79において白抜け領域内の画素 $I(i, j)$ の画素値を計算して埋め込む。画素 $I(i, j)$ の画素値は、次の(2)式によって求める

ことができる。

$$I(i, j) = \{ (B1 - i + B2 - A2) \alpha 1 + (i - A1 + B2 - A2) \beta 1 + (B1 - A1 + B2 - C) \alpha 2 + (B1 - A1 + C - A2) \beta 2 \} / 3 \\ (B1 - A1 + B2 - A2) \dots (2)$$

この式は、ライン方向の開始点A1、終了点B1、カラム方向の開始点A2、終了点B2の4点からの距離に応じた画素値を算出するものである。なお、画素値I(i, j)を定める算出式は(2)式に限らず、白抜け部分の周辺4画素の値と、注目画素から周辺4画素までの距離に依存する算出式であれば任意に定めることができる。

【0060】S80においてi=B1-1か否か、すなわち白抜け領域のライン方向の終了点の手前の画素まで埋め込み処理を終了したか否かを判定し、まだ未処理の画素が残っている場合にはS66へ戻り、同様の処理をi=B1-1となるまで繰り返す。i=B1-1となったらS81へ進む。

【0061】S81では処理ラインの右端まで処理を終えたか否かを判定し、右端まで達していない場合には他の白抜け領域を検出するべくS53に戻り、処理を繰り返す。処理ラインの右端の画素まで処理を終えたら、S82において最終ラインまで処理を終えたか否かを判定し、最終ラインに達していない場合にはS52に戻って同様の処理を最終ラインの処理がすべて終了するまで行なう。最終ラインの処理が終了すると、画像処理部4の処理を終了する。

【0062】図13は、画像処理部の第3の動作例において分離された画像Iの画素値の具体例の説明図、図14は、同じく画像Sの具体例の説明図、図15は、同じく画像Iの一部の埋め込み処理前後の画素値の変化の説明図である。この例では説明を簡単にするために6×8画素で構成される画像を分離した場合を示しており、画像Pは省略した。それぞれ便宜上、1～6のカラム番号と1～8のライン番号を付した。なお、図13、図14において「\*」で示している画素は、説明に関係ないのでその画素値の記載を省略している。また、図15では、図13、図14に一点鎖線で囲んだライン3のカラム2～5の部分のみについて、埋め込み処理前(図15(A))と埋め込み処理後(図15(B))の画素値を示している。

【0063】以下、図10～図12に示したフローチャートに従って具体例における動作を説明してゆく。いま、ライン3が処理ラインであるものとする。S53でi=1となるとS54においてS(1, 3)=0、S(2, 3)=1となり、白抜け領域のライン方向の開始点A1=1、C=3が格納される。

【0064】次にS60でi=5となると、S61においてS(5, 3)=1、S(6, 3)=0となり、白抜け領域のライン方向の終了点B1=6が格納される。

S63においてβ1=I(6, 3)=50、B1=6が格納される。

【0065】S65でi=1とし、S66でi=1+1=2とした後、S67でjを1減じて2とする。S68において、S(2, 2)=0、S(2, 3)=1であるからカラム方向の開始点A2=2が検出され、S70においてα2=I(2, 2)=20、A2=2が格納される。

【0066】続いてS72でj=3-1=2とし、S73で1加算してj=3とした後、S74の条件を判定するが、条件を満足しない。S73でjの値の加算が繰り返され、j=7となるとS(2, 7)=1、S(2, 8)=0となってS74の条件を満足する。すなわちカラム方向の終了点B2=8が検出された。S76においてβ2=I(2, 8)=20、B2=8が格納される。

【0067】このようにして、白抜け領域のライン方向の開始点A1=1、その画素値α1=20、ライン方向の終了点B1=6、その画素値β1=50が得られ、さらにその白抜け領域内の画素(2, 3)について、カラム方向の開始点A2=2、その画素値α2=20、カラム方向の終了点B2=8、その画素値β2=20が得られた。S78においてj=C=3に戻した後、S79において上述の(2)式に従ってI(2, 3)を計算すると、

$$I(2, 3) = \{ (6 - 2 + 8 - 2) \times 20 + (2 - 1 + 8 - 2) \times 50 + (6 - 1 + 8 - 3) \times 20 + (6 - 1 + 3 - 2) \times 20 \} / \{ 3 \times (6 - 1 + 8 - 2) \} = 26$$

となる。この値をI(2, 3)として図15(B)に示すように埋め込む。

【0068】S80でiがライン3の右端まで達していないのでS66に戻り、iに1を加えてi=3とする。i=3のカラムにおいて同様にして白抜け領域のカラム方向の開始点および終了点を求めると、カラム方向の開始点A2=1、その画素値α2=20、カラム方向の終了点B2=5、その画素値β2=90が得られる。

(2)式より、I(3, 3)=44と求められる。同様にしてI(4, 3)=53、I(5, 3)=44と求められ、I(2, 3)～I(5, 3)の値は図15(B)に示すように変換される。

【0069】上述の例ではライン3について説明したが、他のラインについても同様の埋め込み処理が行なわれる。このようにして変換された画像Iは、文字や線画として抽出されて白抜け状態であった画素が、i座標方向だけではなくj座標方向の画素値の変化にも対応した値によって埋められている。そのため、変換後の画像Iでは文字や線画の存在していた部分が2次元的に滑らか

に変化するように埋められており、高周波成分の増加を防いでいる。そのため、例えば2次元離散コサイン変換等の2次元の周波数成分を符号化するような符号化方式を採用して符号化を行なった場合でも、符号量を低減し、復号された画像の品質の低下を防止することができる。

【0070】もちろん、文字や線画として分離されて白抜けとなっている部分が埋められているので、画像Iを低解像度に変換後、元の解像度に再変換して文字や線画と合成しても、文字や線画のエッジ部分に白抜けは発生せず、良好な合成画像を得ることができる。このとき、i座標方向だけではなくj座標方向に対する最近周辺画素の値も参照して変換後の画素値を決めているために、文字部との合成の際にはi座標方向に対しても不具合を生じることはない。

【0071】なお、上述の説明においては、画像分離部2において入力された画像を画像I、画像S、画像Pの3つに分離した。例えばカラー画像であれば例えばR、G、BやC、M、Y等の各色座標系に応じた各色軸毎に3つの画像に分離すればよい。ここで、画像Sは2値の画像、画像Pは濃度を有する画像として文字や線画部分を分離したが、本発明はこれに限らない。画像Sと画像Pを別の画像とせず、濃度を有する1つの文字・線画画像として分離してもよい。この場合、上述の説明において画像Sを参照して値を比較する際には、0かそれ以外の値かを調べればよい。また、さらに入力画像の分離の際に属性データを生成し、生成した属性データも保存するように構成してもよい。

【0072】

【発明の効果】以上の説明から明らかなように、本発明によれば、入力された画像情報を、文字または線画情報からなる第1の画像情報と、文字または線画情報以外の情報からなる第2の画像情報とに分離した際に、第2の画像情報中の第1の画像情報に対応して削除されて白抜けとなった箇所を、当該箇所の周囲の画素を参照して画像情報を埋め込む。これによって、第2の画像情報を低解像度で保持し、画像出力時に元の解像度へ変換し、第1の画像情報と合成した場合でも、イメージ上に書かれた文字の輪郭部分での白抜けはなくなり、良好な合成画像を得ることができる。

【0073】また、イメージ上に細かい文字や線画が多く書かれている場合でも、第2の画像情報中での文字や線画が存在していた領域の白抜けがなくなり、なるべく

平坦な画素値を有する画像となるので、符号化した際に符号量を低減することができ、復号後の画質の低下を抑えることができる。例えばJPGベースライン方式等の離散コサイン変換を用いた可変長且つ非可逆データ圧縮方式を用いた場合でも、圧縮率の低下や、伸長後の画像品質の劣化を防止することが可能となるという効果がある。

【図面の簡単な説明】

【図1】 本発明の画像処理装置の実施の一形態を示すブロック図である。

【図2】 画像データの説明図である。

【図3】 画像処理部の第1の動作例を示すフローチャートである。

【図4】 画像処理部の第1の動作例を示すフローチャート（続き）である。

【図5】 入力画像および分離された各画像の具体例の説明図である。

【図6】 分離された各画像および処理後の画像Iにおける画素値の具体例の説明図である。

【図7】 画像処理部の第2の動作例を示すフローチャートである。

【図8】 画像処理部の第2の動作例を示すフローチャート（続き）である。

【図9】 画像処理部の第2の動作例における画像Iの画素値の具体例の説明図である。

【図10】 画像処理部の第3の動作例を示すフローチャートである。

【図11】 画像処理部の第3の動作例を示すフローチャート（続き）である。

【図12】 画像処理部の第3の動作例を示すフローチャート（続き）である。

【図13】 画像処理部の第3の動作例において分離された画像Iの画素値の具体例の説明図である。

【図14】 画像処理部の第3の動作例において分離された画像Sの具体例の説明図である。

【図15】 画像処理部の第3の動作例において分離された画像Iの一部の埋め込み処理前後の画素値の変化の説明図である。

【図16】 従来の画像処理装置における画像分離処理の具体例の説明図である。

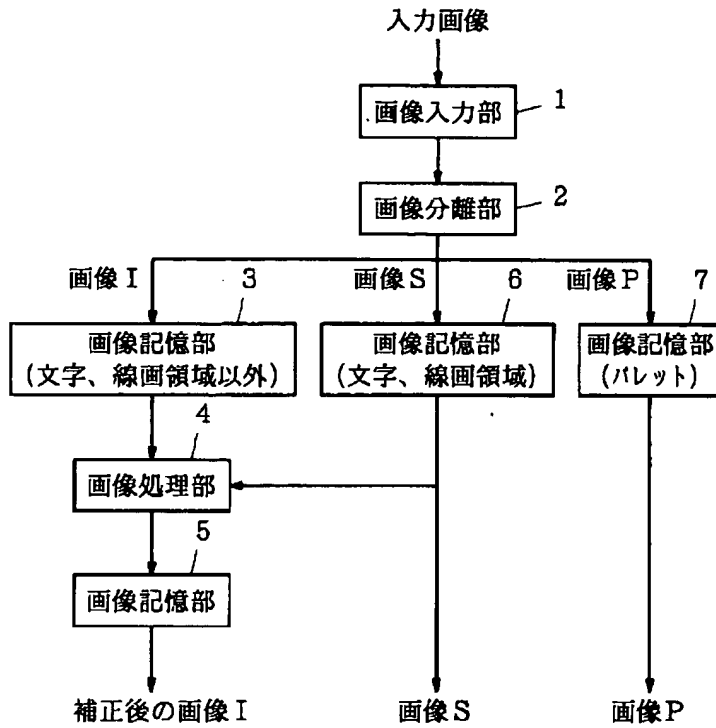
【符号の説明】

1…画像入力部、2…画像分離部、3、5…7…画像記憶部、4…画像処理部。

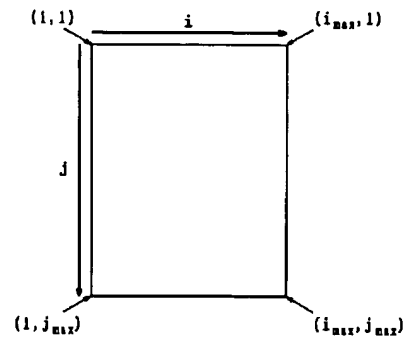
【図15】



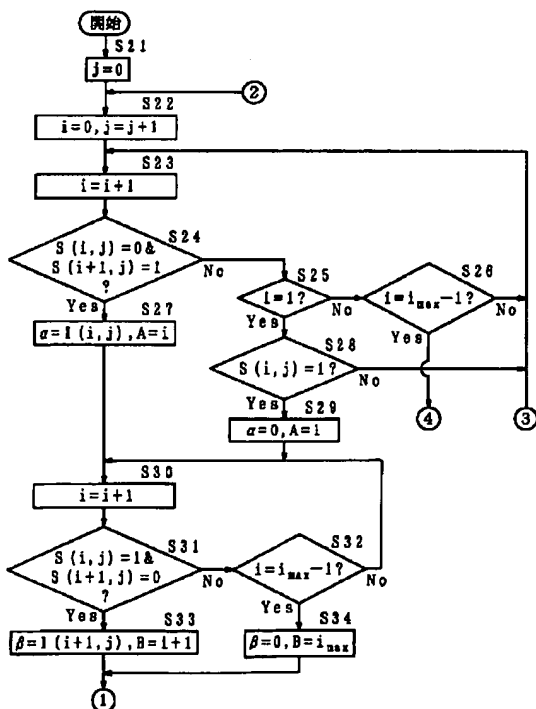
【図1】



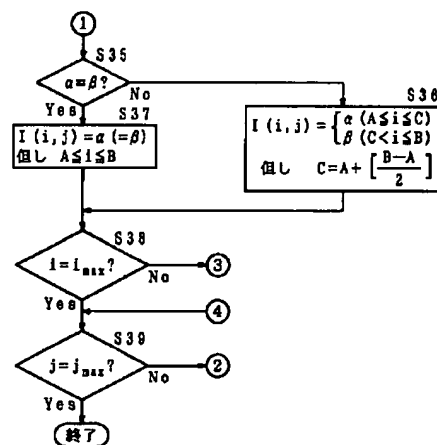
【図2】



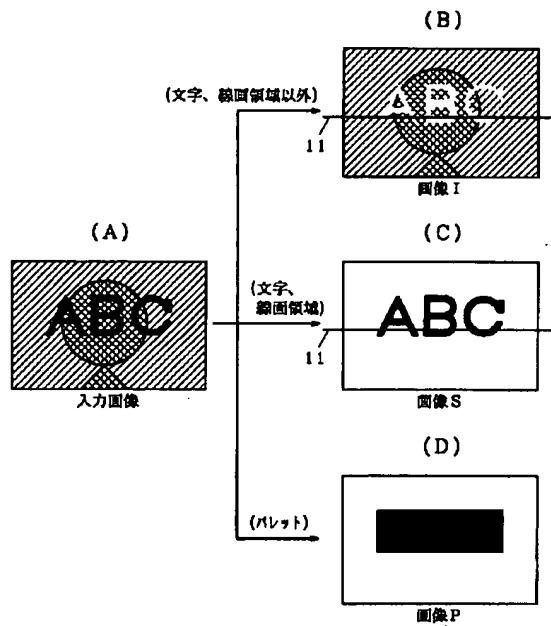
【図3】



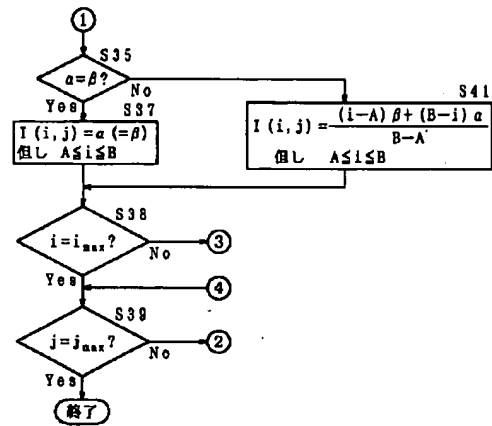
【図4】



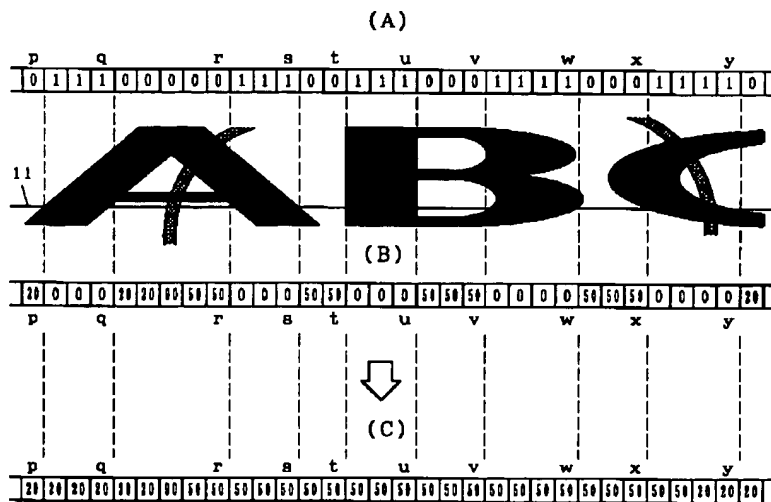
【図5】



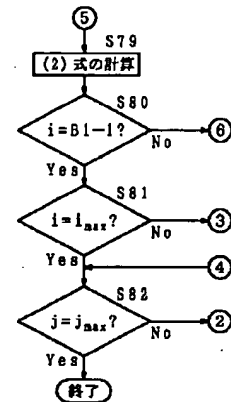
【図8】



【図6】



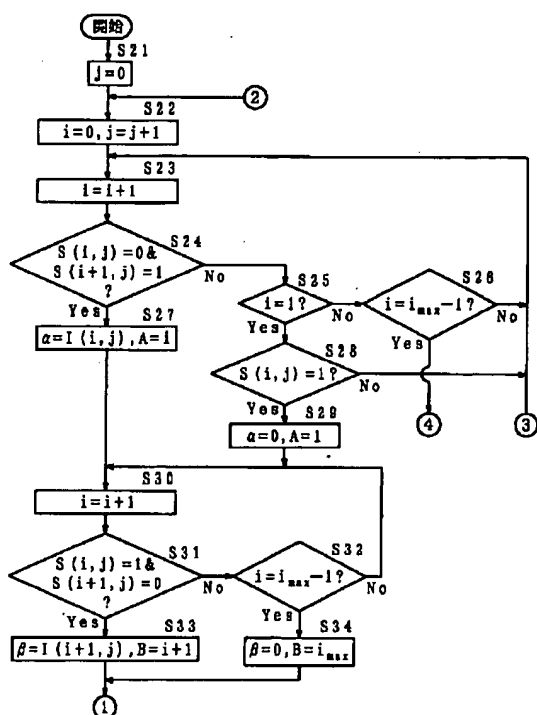
【図12】



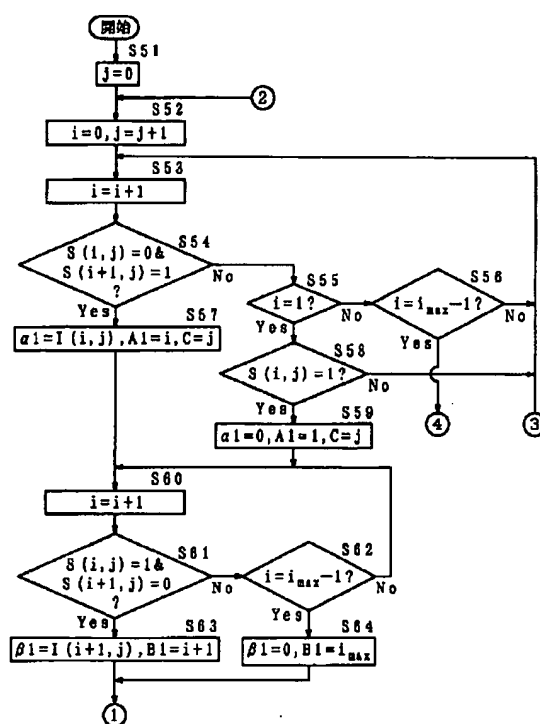
【図13】

i \ j	1	2	3	4	5	6
1	*	*	20	00	*	*
2	*	20	00	00	50	*
3	20	00	00	00	00	50
4	*	00	00	00	00	*
5	*	00	00	60	00	*
6	*	00	*	*	00	*
7	*	00	*	*	00	*
8	*	20	*	*	50	*

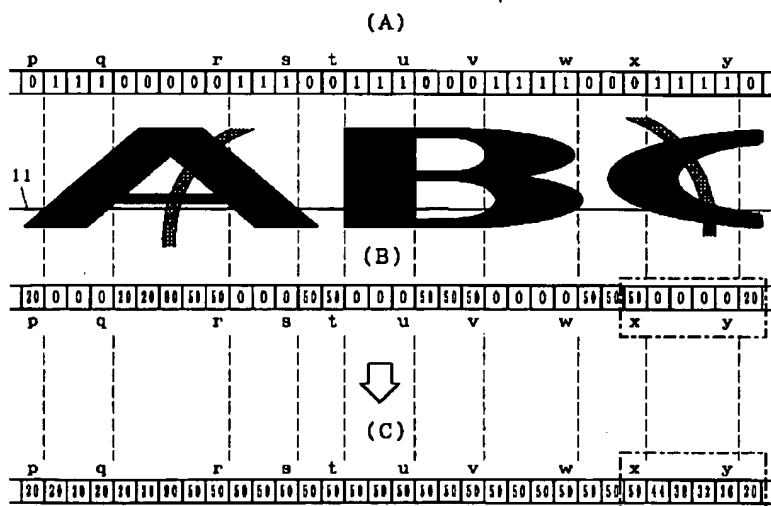
【図7】



【図10】



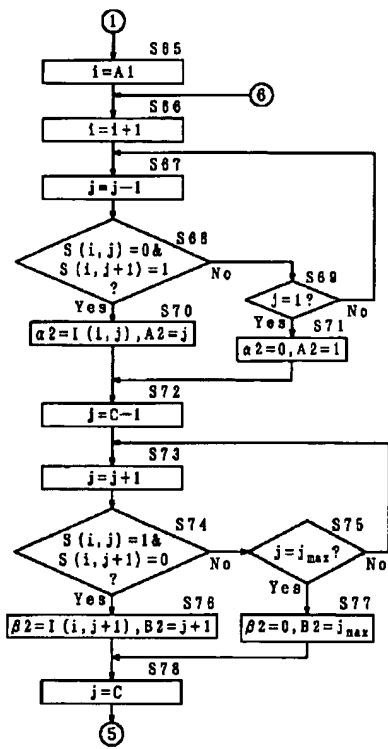
【図9】



【図14】

	1	2	3	4	5	6
1	*	*	0	0	*	*
2	*	0	1	1	0	*
3	0	1	1	1	1	0
4	*	1	1	1	1	*
5	*	1	0	0	1	*
6	*	1	*	*	1	*
7	*	1	*	*	1	*
8	*	0	*	*	0	*

【図11】



【図16】

